

[Descrição](#)

[Público-Alvo/Intended User](#)

[Funcionalidades/Features](#)

[Protótipo de Interfaces do Usuário](#)

[Intro](#)

[Tela Detalhes](#)

[Tela Principal](#)

[Considerações Chave/Key Considerations](#)

[Corner Case.](#)

[Bibliotecas.](#)

[Persistência de dados](#)

[Implementação do Google Play Services.](#)

[Próximos Passos: Tarefas Necessárias](#)

[Tarefa 1: Configuração do Projeto/Project Setup](#)

[Tarefa 2: Implement UI](#)

[Tarefa 3: Implementação de Adapters](#)

[Tarefa 4: Implementação de Holders](#)

[Tarefa 5: Implementação de lógicas](#)

[Tarefa 6: Criação de vetores e adição de imagens](#)

[Tarefa 7: Implementação de testes](#)

Usuário do GitHub: rodrigogaraujo

Party

Descrição

O aplicativo Party utiliza geolocalização para selecionar eventos de acordo com o perfil do usuário em uma determinada distância, não se prendendo apenas a uma cidade, os seus objetivos são dar ao usuário informações necessárias para a compra de ingressos.

Para baladeiros:

- Melhores eventos em sua região
- Melhores eventos na cidade que irá passear
- Descobrir aonde comprar o seu ingresso

Para promotores:

- Divulgação de eventos
- Divulgação de locais de venda de ingresso
- Acesso a quantos usuários visualizaram seu evento

Público-Alvo/Intended User

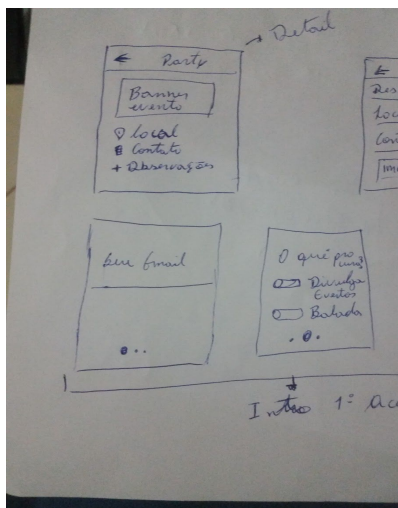
Promotores de eventos e baladeiros de plantão.

Funcionalidades/Features

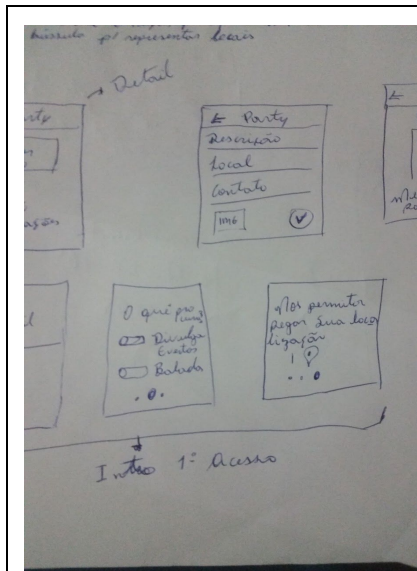
- Listagem de anúncios
- Adicionar anúncios
 1. Adicionar foto do evento
 2. Adicionar locais de venda
- Listagem de bares, restaurantes e clubes em determinada região
- Possibilidade de uso do aplicativo estando offline

Protótipo de Interfaces do Usuário

Intro



A introdução, que será desenvolvida com um ViewPager somando com alguns fragmentos, assim o App pegará as informações necessárias dos usuários, sabendo assim em que mais estarão interessados e realizar a lógica do aplicativo após realizarem seus cadastros.

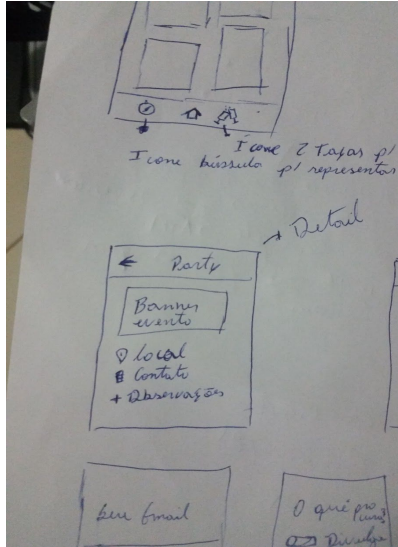


Tela principal



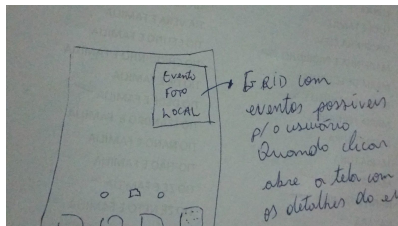
A tela principal, possui um GridView para a listagem de eventos, Um Menu Drawer para o usuários ter acesso as configurações básicas como: Atualização de local e criação de eventos (Caso seja um promotor), em baixo terá 3 opções: Home - para voltar ao início, Locais - para a listagem de locais cadastrados (promotores ao cadastrar um evento automaticamente estarão criando um local), Eventos - onde serão listados todos os eventos disponíveis no banco de dados, sem nenhum filtro principal

Tela detalhes



Aqui teremos os detalhes de um evento, com todas as informações necessárias aos baladeiros e na parte de cima um toolbar para o usuário poder voltar a tela principal

Widget



O widget terá um grid contendo alguns eventos que o usuário poderá se interessar, e ao clicar em algum evento o aplicativo irá abrir a tela de detalhes de eventos com o item selecionado pelo usuário

Considerações Chave/Key Considerations

O aplicativo utilizará o Firebase Database como banco de dados pois é um banco de dados NoSql e em tempo real, facilitando assim a vida do usuário, o Firebase Auth para as autenticações e criações de novos usuários e SharedPreferences para enviar itens que o usuário poderá se interessar para o widget.

Corner Case

Ao entrar no aplicativo pela primeira vez, o usuário terá que se cadastrar no Party. Tendo duas opções de usuários, promotor de eventos ou baladeiros. Feito isso caso o usuário seja apenas baladeiro, a lógica irá tratar para que a opção de Criação de eventos não esteja disponível e a opção ficará visível caso seja um anunciante.

Então teremos:

Caso o usuário não tenha conexão à Internet o aplicativo irá notificar o usuário com um mensagem de erro como um Snack ou um TextView que aparece apenas nessa ocasião.

Caso promotor teremos em sua tela principal eventos que estão em sua proximidade (padrão dez quilômetros) e um FAB para ele adicionar um novo evento. Ao clicar no FAB irá ser redirecionado para um tela onde irá cadastrar um novo evento com as seguintes informações: Foto do evento, possíveis cantores ou DJ's, de acordo com o evento caso haja, contato, local e locais para a compra de ingressos.

Caso baladeiro a tela principal mostra de padrão eventos em até dez quilômetros (o usuário poderá alterar em uma tela de profile) de distância do ponto atual do usuário, caso ele clique em algum evento, abre uma tela de detalhes do evento, tendo assim a localização o usuário poderá clicar e ser redirecionado para uma tela onde poderá traçar a rota para o local do evento, e com informações de locais para a compra de ingressos.

Bibliotecas

Picasso para o carregamento e cacheamento de imagens.

ButterKnife para facilitar o acesso as views.

Espresso e JUnit para testes no aplicativo.

Persistência de dados

A princípio utilizarei o Firebase Database e o Firebase Auth para facilitar o acesso a dados, o Database para operações como por exemplo cadastro de eventos, atualização de eventos e o Auth facilita na autenticação e cadastro do usuário, com o SDK do Firebase Auth já posso importar ao meu projeto telas de login que já possuem opções de cadastro de usuário pois ao informar o email o próprio firebase irá saber se aquele email já está cadastrado ou não. Assim deixo o meu código mais limpo e prático. Pois o Firebase é em realtime ajudando assim no processo de CRUD do aplicativo. Pois os mesmos me permitem não perder tempo com criação de um Webservice para um app desse porte e o Firebase faz operações do CRUD ao database com poucas linhas de código, ficando muito mais prático o seu manuseio.

Implementação do Google Play Services

Firebase Database e o Firebase Auth.

Próximos Passos: Tarefas Necessárias

Tarefa 1: Configuração do Projeto/Project Setup

- Configurar bibliotecas
- Criar um aplicativo no console firebase
- Adicionar a Key do firebase ao projeto

Tarefa 2: Implementar a Interface de Usuário (UI)

- Construir a UI da MainActivity
- Construir a UI para alguma outra coisa
- Construir a UI para listar os itens
- Construir uma Intro com ViewPager e Fragments para pegar os dados do usuário
- Construir a UI da MainActivity
- Construir a Fragment para a listagem de Locais na MainActivity
- Construir uma Activity para o cadastro de Locais
- Construir um layout para listar no Fragment de Locais
- Construir a Fragment para a listagem de Eventos na MainActivity
- Construir uma Activity para o cadastro de Eventos
- Construir um layout para listar no Fragment de eventos

Tarefa 3: Implementar os Adapters

- Criação de Adapter para Locais
- Criação de Adapter para Eventos

Tarefa 4: Implementar os Holders

- Criação de Holder para Locais
- Criação de Holder para Eventos

Tarefa 5: Implementar lógicas

- Implementar a lógica para cadastro de novos usuários
- Implementar a lógica de login de usuários
- Implementar o cadastro de Locais
- Implementar a listagem de Locais
- Implementar a cadastro de Eventos
- Implementar a listagem de Eventos

Tarefa 6: Criação de vetores e adição de imagens

- Criar vetores e adicionar imagens para ações como: confirmar, cancelar, etc.

Tarefa 7: Implementação de test

- Criação de testes unitários para verificar se o aplicativo contém bugs.
-